

# A Method for Smooth Merging of Electron Density Distributions at the Chromosphere-Corona Boundary

Leonid Benkevitch and Divya Oberoi  
MIT Haystack Observatory, Westford MA

## ABSTRACT

The electron number density ( $N_e$ ) distributions in solar chromosphere and corona are usually described with models of different nature: exponential for the former and inverse power law for the latter. Moreover, the model functions often have different dimensionality, e.g. the chromospheric distribution may depend solely on solar altitude, while the coronal number density may be a function of both altitude and latitude. For applications which need to consider both chromospheric and coronal models, the chromosphere-corona boundary, where these functions have different values as well as gradients, can lead to numerical problems. We encountered this problem in context of ray tracing through the corona at low radio frequencies, as a part of effort to prepare for the analysis of solar images from new generation radio arrays like the Murchison Widefield Array (MWA), Low Frequency Array (LOFAR) and Long Wavelength Array (LWA). We have developed a solution to this problem by using a *patch* function, a thin layer between the chromosphere and the corona which matches the values and gradients of the two regions at their respective interfaces. We describe the method we have developed for defining this patch function to seamlessly “stitch” chromospheric and coronal electron density distributions, and generalize the approach to work for any arbitrary distributions of different dimensionality. We show that the complexity of the patch function is independent of the stitched functions dimensionalities. It always has eight parameters (even four for univariate functions) and they may be found without linear system solution for every point. The developed method can potentially be useful for other applications.

## 1. Introduction

The visible surface of the sun, the photosphere, has the radius  $1R_\odot \approx 6.955 \times 10^5$  km. The photosphere is surrounded by the solar atmosphere, which consists of two major layers. The lower layer, the chromosphere, is relatively thin. It extends up to several thousand kilometers above the photosphere. The upper layer, the corona, is a gaseous envelope with the density rapidly decreasing with the radius. Depending upon the nature of the study, a working number for size of the corona can be determined based on the coronal height where the coronal plasma becomes too tenuous to be discernible for intended purposes. The size of the corona might thus be considered to range from tens up to hundreds of the solar radii.

Electromagnetic rays at low frequencies in the

heliosphere propagate in the regions with the plasma density lower than the critical density  $\rho_{cr}$ , defined as

$$\rho_{cr} = \frac{m_p m_e \omega^2}{4\pi e^2}, \quad (1)$$

where  $e$  is the electron charge,  $m_e$  is the electron mass,  $m_p$  is the proton mass, and  $\omega$  is the radio wave frequency in  $\text{rad s}^{-1}$ . At the critical plasma density the dielectric permittivity  $\epsilon$  of the plasma and, hence, its index of refraction for a given frequency  $\omega$  become zero. This implies that for most coronal models, radiation at frequencies  $< 150\text{--}180$  MHz does not penetrate to the chromosphere and travels only in the corona. At higher frequencies, the rays with sharp angles of incidence can penetrate into the much denser and cooler chromosphere. Modeling of propagation of radio frequency radiation through the so-

lar atmosphere hence requires knowledge of both chromospheric and coronal electron density ( $N_e$ ) distributions. Our method of choice for ray tracing (Benkevitch et al., 2010), a fast, second order algorithm, requires smoothness in both  $N_e$  and its gradient ( $\nabla N_e$ ) everywhere in the medium. The fact that chromospheric and coronal density distributions come from independent models which are not consistent at the boundary between them leads to a discontinuity in  $N_e$  and  $\nabla N_e$  at this boundary. Though the transition between the chromosphere and the corona is intrinsically very rapid and involves rather large density gradients, the discontinuity in  $N_e$  and  $\nabla N_e$  is unphysical and leads to numerical instability in the ray tracing algorithm. It is therefore essential to make these distributions consistent at their interface and get them to smoothly morph from one to another while maintaining a smooth gradient. An attendant problem is the different dimensionality of the chromospheric and coronal density distributions: the former is one-dimensional, depending only on the solar radius, while the latter can be two-dimensional, depending both on the radius and (co)latitude.

In this work we offer our solution to this problem. We use the concept of a thin *patch* region which encloses the chromosphere/corona interface, and which is consistent with the chromospheric and coronal models in either region. The *patch* is a function constructed to allow the two regions to smoothly morph into each other and accommodates the differences in their dimensionalities. In Section 2 we present the plasma density models and a mathematical formulation of the problem. In Section 3 we start with the simplest case of stitching two univariate functions. The method is developed further for the case of merging of a univariate and a multivariate functions in Section 4. The method appears to be universal enough to extend it to the problem of merging two multivariate functions along one dimension. The appropriate theory is developed in Section 5. Numerical implementation details are discussed in Section 6, in particular it is shown that the suggested merging method is not computationally intensive.

We came across this problem while building the formalism to implement ray tracing through the solar atmosphere at low radio frequencies. Our work is motivated by the expected near term avail-

ability of high fidelity solar images from the new generation radio arrays like the Murchison Wide-field Array (MWA) (Lonsdale et al., 2009). The early results from the 32 element MWA engineering prototype are very promising and currently represent the state of the art in high fidelity solar imaging at these frequencies (Oberoi et al., 2011). This work is of direct relevance for pursuing solar science with other new generation arrays like the Low Frequency Array (LOFAR) (de Vos et al., 2009) and Long Wavelength Array (LWA) (Ellingson et al., 2009). We hope this formulation is general enough to be applicable to similar problems in other independent contexts.

## 2. Formulation

There are a number of models describing the electron number density  $N_e(\text{cm}^{-3})$  in the chromosphere and the corona. For the chromosphere we use the model by Cillie and Menzel (Cillie & Menzel 1935):

$$N_e(r) = 5.7 \times 10^{11} e^{-7.7 \times 10^{-4}(R_\odot(r-1)-500)}, \quad (2)$$

where  $r$  is the distance from the center of sun measured in the solar radii,  $R_\odot$ . This model is applicable for a chromosphere with a thickness of about 10,000 km, or  $0.014378R_\odot$ . The simplest available coronal models are one dimensional in nature, depending only on the solar distance, e.g. the Newkirk model (Newkirk 1961):

$$N_e(r) = 4.2 \times 10^{4+4.32/r}, \quad (3)$$

and the Baumbach and Allen model (Allen 1947):

$$N_e(r) = 10^8(1.55r^{-6} + 2.99r^{-16}). \quad (4)$$

where  $r$  is the solar distance in units of  $R_\odot$ .

However, observations show that the plasma density in the corona falls faster the solar distance in the polar regions than it does in equatorial region, especially close to solar minima. This gave rise to development of more complex models. A popular one is by Kuniiji Saito (Saito 1970). This coronal density model depends on two variables, the spherical coordinate  $r$  and  $\theta$ , representing the radial distance from the center of the Sun and the

colatitude, respectively:

$$N_e(r, \theta) = 3.09 \times 10^8 r^{-16} (1 - 0.5 \cos \theta) + 1.56 \times 10^8 r^{-6} (1 - 0.95 \cos \theta) + 0.0251 \times 10^8 r^{-2.5} (1 - \sqrt{\cos \theta}). \quad (5)$$

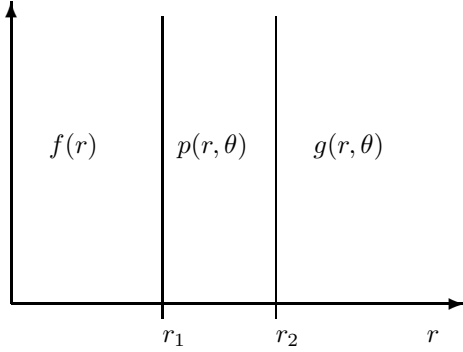


Fig. 1.— Two functions  $f(r)$  and  $g(r, \theta)$ , defined over different parts of space, are stitched together by a third, patch function  $p(r, \theta)$  in the intermediate space segment between  $r = r_1$  and  $r = r_2$ .

Consider a problem of stitching together the chromospheric and coronal models described in Eqs. 3 and 5, respectively. Consider a relatively thin transition layer extending from  $r_1$  to  $r_2$ , where  $r_1$  is at the top of the chromosphere, and  $r_2$  is at the bottom of the corona. For this specific problem the limits can be chosen as  $r_1 = 9,000$  km and  $r_2 = 11,000$  km above the photosphere. This transition layer encloses the chromosphere-corona boundary. Denote the  $N_e$  distribution functions in the chromosphere and the corona by  $f(r)$  and  $g(r, \theta)$ , respectively, as shown in Fig. 1. The *patch* function,  $p(r, \theta)$ , defined in the transition layer between  $r_1$  and  $r_2$ , must match  $f(r)$  and  $g(r, \theta)$  at both its ends in both its value and derivatives. Finding an appropriate  $p(r, \theta)$  requires (1) choosing a family of functions for  $p(r, \theta)$  and (2) finding its parameters to satisfy the boundary conditions for smoothness.

Note that  $f(r)$  and  $g(r, \theta)$  have different number of independent variables. Our approach can adequately accommodate this complication,

though we start by considering a simpler case of stitching two one-dimensional distributions.

### 3. Merging one-variable functions

Consider the case where both chromospheric and coronal electron number density distributions are only dependent on one variable, the radial distance  $r$ . As shown schematically in Fig. 1,  $r_1 < r_2$ ,  $f(r)$  on the left is the Cillie and Menzel distribution (2), and  $g(r)$  on the right is either Newkirk or Baumbach distribution. We want the values of  $p(r)$  at  $r_1$  and  $r_2$  to be equal to those of the distributions on the left and right ends, respectively. Also, we want the derivative of  $p(r)$ , denoted as  $p'(r)$ , to take boundary values equal those of the derivatives of the distributions,  $f'(r)$  and  $g'(r)$  on the left and on the right, respectively. Putting the boundary conditions together in one system:

$$\begin{cases} p(r_1) = f(r_1) \\ p(r_2) = g(r_2) \\ p'(r_1) = f'(r_1) \\ p'(r_2) = g'(r_2) \end{cases} \quad (6)$$

We have four equations, therefore  $p(r)$  must have exactly four unknown parameters to be fully determined through solving system (6). We also would like to have the simplest system to solve, i.e. a linear system. These considerations lead to the choice of a third order polynomial as  $p(r)$ ,

$$p(r) = a_0 + a_1 r + a_2 r^2 + a_3 r^3. \quad (7)$$

Substitution of the polynomial (7) into the system (6) yields the linear system

$$\begin{cases} a_0 + a_1 r_1 + a_2 r_1^2 + a_3 r_1^3 = f(r_1) \\ a_0 + a_1 r_2 + a_2 r_2^2 + a_3 r_2^3 = g(r_2) \\ a_1 + 2a_2 r_1 + 3a_3 r_1^2 = f'(r_1) \\ a_1 + 2a_2 r_2 + 3a_3 r_2^2 = g'(r_2) \end{cases} \quad (8)$$

Its solution, the vector  $\mathbf{a} = \|a_0; a_1; a_2; a_3\|^T$ , comprises the coefficients of polynomial (7). Once these coefficients have been determined, the patch polynomial (7), by construction, has value and derivatives equal to those of the chromospheric distribution  $f(r)$  at its left end and the coronal distribution  $g(r)$  at its right. The system (8) can be

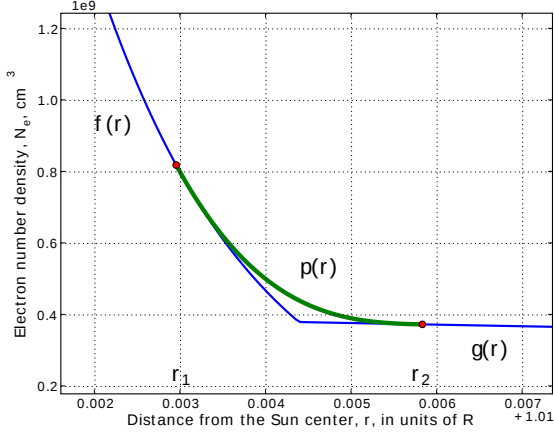


Fig. 2.— Stitching of the two functions  $f(r)$  and  $g(r)$ , by a patch function  $p(r)$  between  $r = r_1$  and  $r = r_2$ .

also written compactly in the matrix-vector form,

$$\mathbf{P}\mathbf{a} = \mathbf{c}, \quad (9)$$

where  $\mathbf{c} = \|f(r_1); g(r_2); f'(r_1); g'(r_2)\|^T$  is the right-hand-side vector, and  $\mathbf{P}$  is the system matrix,

$$\mathbf{P} = \begin{pmatrix} 1 & r_1 & r_1^2 & r_1^3 \\ 1 & r_2 & r_2^2 & r_2^3 \\ 0 & 1 & 2r_1 & 3r_1^2 \\ 0 & 1 & 2r_2 & 3r_2^2 \end{pmatrix} \quad (10)$$

An example of smooth stitching of the chromospheric and coronal  $N_e$  distributions, given by equations (2) and (4), is given in Fig. 2.

#### 4. Merging one- and multi-variable functions along one dimension

We now generalize the solution to the case with a function of one variable at  $r_1$  and a function of two or more variables at  $r_2$ . A real-world example here is the problem of merging the model of Cillie and Menzel (Cillie & Menzel 1935), Eq. (2) in the chromosphere with the model of Saito (1970), Eq. (5) in the corona. The patch  $p(r, \theta)$  is now a function of two variables. It must smoothly match both values and derivatives of  $f(r)$  at its left boundary and of  $g(r, \theta)$  at the right boundary. Note that  $f(r)$  has only one derivative, while  $g(r, \theta)$  has two partial derivatives. In order to guess the kind of a patch function we count the

number of constraints (or degrees of freedom) imposed on the patch: two on the end point values and four on the partial derivatives at the end points, or six on total. For a completely asymmetric coronal distribution  $g(r, \theta, \varphi)$ , when it is a function of all the three variables, the radial distance  $r$ , the colatitude  $\theta$ , and the longitude  $\varphi$ , the number of conditions imposed on the patch  $p(r, \theta, \varphi)$  becomes  $2 + 3 \times 2 = 8$ . These calculations suggest that the complexity of the patch function apparently grows with the dimensionality of the stitched functions. However, this is not the case. As we shall show here and in the next Section, the patch function can be build such that it always depends only on eight parameters independent of the dimensionality of the stitched functions. The method can be used for stitching functions of arbitrary (and different) numbers of dimensions, along one of them.

Note that the patch function must gradually turn from multi-variable  $g(r, \theta, \varphi)$  at  $r_2$  into single-variable  $f(r)$  at  $r_1$ . Consider a function  $g(r_2, \theta, \varphi)$ . It has its variable  $r$  “frozen” at  $r_2$ , so that  $g(r_2, \theta, \varphi)$  is now dependent only on the two remaining variables,  $\theta$  and  $\varphi$ . Form the product of the  $g(r_2, \theta, \varphi)$  and some polynomial  $b(r)$ , which has value of 0 at  $r_1$  and 1 at  $r_2$ . The function  $b(r)g(r_2, \theta, \varphi)$  is not only equal to 0 at  $r_1$ , its partial derivatives with respect to  $\theta$  and  $\varphi$  also evaluate to 0 at  $r_1$ , i.e. at  $r_1$  it effectively depends on  $r$  only. However, it cannot be linked to  $f(r)$  directly because generally  $f(r_1) \neq 0$ . To make such linkage possible, we add one more polynomial,  $a(r)$ , so the patch function takes the form

$$p(r, \theta, \varphi) = a(r) + b(r)g(r_2, \theta, \varphi). \quad (11)$$

The polynomial  $a(r)$  must be equal  $f(r)$  at  $r_1$ , and must be equal zero at  $r_2$ , where  $b(r_2)g(r_2, \theta, \varphi)$  equals  $g(r, \theta, \varphi)$ . Thus, the conditions imposed on the polynomials are

$$\begin{aligned} a(r_1) &= f(r_1); & a(r_2) &= 0; \\ b(r_1) &= 0; & b(r_2) &= 1. \end{aligned} \quad (12)$$

For the multi-variable functions the boundary conditions include their partial derivatives. Listing the boundary conditions as a system of equations

yields:

$$\left\{ \begin{array}{l} p(r_1, \theta, \varphi) = a(r_1) + b(r_1)g(r_2, \theta, \varphi) = f(r_1) \\ p_r(r_1, \theta, \varphi) = a_r(r_1) + b_r(r_1)g(r_2, \theta, \varphi) = f_r(r_1) \\ p_\theta(r_1, \theta, \varphi) = b(r_1)g_\theta(r_2, \theta, \varphi) = 0 \\ p_\varphi(r_1, \theta, \varphi) = b(r_1)g_\varphi(r_2, \theta, \varphi) = 0 \\ p(r_2, \theta, \varphi) = a(r_2) + b(r_2)g(r_2, \theta, \varphi) \\ \quad = g(r_2, \theta, \varphi) \\ p_r(r_2, \theta, \varphi) = a_r(r_2) + b_r(r_2)g(r_2, \theta, \varphi) \\ \quad = g_r(r_2, \theta, \varphi) \\ p_\theta(r_2, \theta, \varphi) = b(r_2)g_\theta(r_2, \theta, \varphi) = g_\theta(r_2, \theta, \varphi) \\ p_\varphi(r_2, \theta, \varphi) = b(r_2)g_\varphi(r_2, \theta, \varphi) = g_\varphi(r_2, \theta, \varphi) \end{array} \right. \quad (13)$$

Here we denote a partial derivative with respect to a variable by putting this variable at the subscript position after the function name, e.g.  $g_r(r, \theta, \varphi)$  is the derivative of  $g(r, \theta, \varphi)$  with respect to  $r$ . Now we simplify the system (13) using conditions (12) on the polynomials  $a(r)$  and  $b(r)$ :

$$\left\{ \begin{array}{l} p(r_1, \theta, \varphi) = a(r_1) = f(r_1) \\ p_r(r_1, \theta, \varphi) = a_r(r_1) + b_r(r_1)g(r_2, \theta, \varphi) = f_r(r_1) \\ p_\theta(r_1, \theta, \varphi) = 0 = 0 \\ p_\varphi(r_1, \theta, \varphi) = 0 = 0 \\ p(r_2, \theta, \varphi) = g(r_2, \theta, \varphi) = g(r_2, \theta, \varphi) \\ p_r(r_2, \theta, \varphi) = a_r(r_2) + b_r(r_2)g(r_2, \theta, \varphi) \\ \quad = g_r(r_2, \theta, \varphi) \\ p_\theta(r_2, \theta, \varphi) = g_\theta(r_2, \theta, \varphi) = g_\theta(r_2, \theta, \varphi) \\ p_\varphi(r_2, \theta, \varphi) = g_\varphi(r_2, \theta, \varphi) = g_\varphi(r_2, \theta, \varphi) \end{array} \right. \quad (14)$$

We can see that the first, the third, the fourth, the fifth, the seventh, and the eighth equations in (14) are nothing but tautologies and can be removed from the system, which now takes the form

$$\left\{ \begin{array}{l} a_r(r_1) + b_r(r_1)g(r_2, \theta, \varphi) = f_r(r_1) \\ \quad = p_r(r_1, \theta, \varphi) \\ a_r(r_2) + b_r(r_2)g(r_2, \theta, \varphi) = g_r(r_2, \theta, \varphi) \\ \quad = p_r(r_2, \theta, \varphi) \end{array} \right. \quad (15)$$

If we for simplicity fix the  $a(r)$  derivatives at zero, this set of equations provides boundary conditions

for the derivatives of  $b(r)$ , :

$$\left\{ \begin{array}{l} a_r(r_1) = 0 \\ a_r(r_2) = 0 \\ b_r(r_1) = \frac{f_r(r_1)}{g(r_2, \theta, \varphi)} \\ b_r(r_2) = \frac{g_r(r_2, \theta, \varphi)}{g(r_2, \theta, \varphi)}. \end{array} \right. \quad (16)$$

We remember that there are four constraints imposed on each of the polynomials: two on their end-point values and two on their partial derivatives with respect to the stitching-dimension variable  $r$ . Therefore, both polynomials are of the  $3^{rd}$  order, have four unknown coefficients each, and require eight equations to be solved for. We have four equations in (16); the other four are in (12). Combining the both provides two linear systems that can be used to determine coefficients of the polynomials  $a(r)$  and  $b(r)$ :

$$\left\{ \begin{array}{l} a(r_1) = f(r_1) \\ a(r_2) = 0 \\ a_r(r_1) = 0 \\ a_r(r_2) = 0 \end{array} \right. \quad (17)$$

and

$$\left\{ \begin{array}{l} b(r_1) = 0 \\ b(r_2) = 1 \\ b_r(r_1) = \frac{f_r(r_1)}{g(r_2, \theta, \varphi)} \\ b_r(r_2) = \frac{g_r(r_2, \theta, \varphi)}{g(r_2, \theta, \varphi)}. \end{array} \right. \quad (18)$$

One can notice that both linear systems have a nice common property, they use the same system matrix  $\mathbf{P}$  as used in the previous Section (see Eq. (8) through (10)). The systems (17) and (18) can be rewritten in the vector-matrix form as follows:

$$\mathbf{P}\mathbf{a} = \mathbf{c}_a, \quad (19)$$

$$\mathbf{P}\mathbf{b} = \mathbf{c}_b, \quad (20)$$

where  $\mathbf{a} = \|a_0; a_1; a_2; a_3\|^T$  and  $\mathbf{b} = \|b_0; b_1; b_2; b_3\|^T$  are vectors of coefficients for the polynomials

$$a(r) = a_0 + a_1r + a_2r^2 + a_3r^3. \quad (21)$$

$$b(r) = b_0 + b_1r + b_2r^2 + b_3r^3. \quad (22)$$

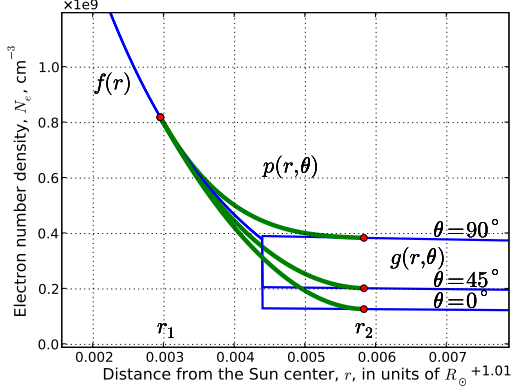


Fig. 3.— Stitching of the  $f(r)$  function of a single variable and the function  $g(r, \theta)$  of two independent variables along the  $r$  dimension between  $r = r_1$  and  $r = r_2$ . The patch function  $p(r, \theta)$  has the form given by Eq. (11). The  $g(r, \theta)$  function is plotted for three values of colatitude  $\theta$ ,  $0^\circ$ ,  $45^\circ$ , and  $90^\circ$ .

The right-hand-side vectors  $\mathbf{c}_a$  and  $\mathbf{c}_b$  are determined according to (17) and (18):

$$\mathbf{c}_a = \parallel f(r_1); 0; 0; 0 \parallel^T, \quad (23)$$

$$\mathbf{c}_b = \parallel 0; 1; \frac{f_r(r_1)}{g(r_2, \theta, \varphi)}; \frac{g_r(r_2, \theta, \varphi)}{g(r_2, \theta, \varphi)} \parallel^T. \quad (24)$$

Substitution of  $a(r)$  and  $b(r)$  in (11) gives a smooth patch function to link together the  $f()$  and  $g()$  functions.

In order to test the method we stitch two distributions dependent on different number of variables, those by Cillie and Menzel's (Cillie & Menzel 1935), Eq. (2) and by Saito (1970), Eq. (5). The former depends only on  $r$ , while the latter depends on  $r$  and  $\theta$ . The stitching occurs along the  $r$  coordinate. Fig. 3 shows three stitching cases for different colatitudes  $\theta$ . This example illustrates the application of the general method described here.

## 5. Merging two multivariable functions along one dimension

It is possible to generalize the one-and-multi-variable merging method to stitch together two multi-variable functions along the axis of one of the variables they have in common. For example, we may need to smoothly merge two functions,

$f(r, \theta, \varphi)$  and  $g(r, \theta, \varphi)$  along the  $r$  dimension using the patch function  $p(r, \theta, \varphi)$  over the interval  $[r_1, r_2]$ . The patch function must have the properties of  $f(r, \theta, \varphi)$  (value and derivatives) at the  $r_1$  end, but on the way from  $r_1$  to  $r_2$  these properties must smoothly morph into being indistinguishable from  $g(r, \theta, \varphi)$ . To build such a function, consider two functions formed from  $f$  and  $g$  by “freezing” the  $r$  variable:  $f(r_1, \theta, \varphi)$  and  $g(r_2, \theta, \varphi)$ . Following the method developed in the previous Section, we introduce two polynomials,  $a(r)$  and  $b(r)$ . In the interval  $[r_1, r_2]$ ,  $a(r)$  should drop off from one to zero, while  $b(r)$  should increase from zero to one. Then the function:

$$p(r, \theta, \varphi) = a(r)f(r_1, \theta, \varphi) + b(r)g(r_2, \theta, \varphi) \quad (25)$$

will behave as desired to serve as a valid patch function. The boundary conditions imposed on the values of the polynomials  $a(r)$  and  $b(r)$  are:

$$\begin{aligned} a(r_1) &= 1; & a(r_2) &= 0; \\ b(r_1) &= 0; & b(r_2) &= 1. \end{aligned} \quad (26)$$

As before, we build a system of equations comprising all the boundary conditions on  $p(r, \theta, \varphi)$  at the end points  $r_1$  and  $r_2$  similar to system (13): the values and derivatives of  $p(r, \theta, \varphi)$  are equated to those of  $f(r, \theta, \varphi)$  on the left and those of  $g(r, \theta, \varphi)$  on the right. After removing the tautologies only two equations remain:

$$\begin{cases} a_r(r_1)f(r_1, \theta, \varphi) + b_r(r_1)g(r_2, \theta, \varphi) = f_r(r_1, \theta, \varphi) \\ \phantom{a_r(r_1)f(r_1, \theta, \varphi) + b_r(r_1)g(r_2, \theta, \varphi)} = p_r(r_1, \theta, \varphi) \\ a_r(r_2)f(r_1, \theta, \varphi) + b_r(r_2)g(r_2, \theta, \varphi) = g_r(r_2, \theta, \varphi) \\ \phantom{a_r(r_2)f(r_1, \theta, \varphi) + b_r(r_2)g(r_2, \theta, \varphi)} = p_r(r_2, \theta, \varphi) \end{cases} \quad (27)$$

This system of two equations has four unknowns, the derivatives of polynomials  $a(r)$  and  $b(r)$  at both ends of the interval  $[r_1, r_2]$ . For simplicity, we assign values of zero to  $a_r(r_2)$  and  $b_r(r_1)$  to form the system

$$\begin{cases} a_r(r_1) = \frac{f_r(r_1, \theta, \varphi)}{f(r_1, \theta, \varphi)} \\ a_r(r_2) = 0 \\ b_r(r_1) = 0 \\ b_r(r_2) = \frac{g_r(r_2, \theta, \varphi)}{g(r_2, \theta, \varphi)}. \end{cases} \quad (28)$$

Combining conditions on the polynomials  $a(r)$  and  $b(r)$  from (26) and (28) we build two linear systems:

$$\begin{cases} a(r_1) = 1 \\ a(r_2) = 0 \\ a_r(r_1) = \frac{f_r(r_1, \theta, \varphi)}{f(r_1, \theta, \varphi)} \\ a_r(r_2) = 0 \end{cases} \quad (29)$$

and

$$\begin{cases} b(r_1) = 0 \\ b(r_2) = 1 \\ b_r(r_1) = 0 \\ b_r(r_2) = \frac{g_r(r_2, \theta, \varphi)}{g(r_2, \theta, \varphi)} \end{cases} \quad (30)$$

There are four constraints imposed on each of the polynomials, two on the end-point values and two on the partial derivatives with respect to  $r$ . Both polynomials are cubic, like (21) and (22). We already have eight equations to solve for the eight coefficients of  $a(r)$  and  $b(r)$ . Both system have the same system matrix  $\mathbf{P}$ , given in (10), and can be compactly written in the matrix-vector form as shown in (19) and (20). The right-hand-side vectors for systems (29) and (30) are

$$\mathbf{c}_a = \begin{bmatrix} 1; & 0; & \frac{f_r(r_1, \theta, \varphi)}{f(r_1, \theta, \varphi)}; & 0 \end{bmatrix}^T, \quad (31)$$

$$\mathbf{c}_b = \begin{bmatrix} 0; & 1; & 0; & \frac{g_r(r_2, \theta, \varphi)}{g(r_2, \theta, \varphi)} \end{bmatrix}^T. \quad (32)$$

Thus  $a(r)$  and  $b(r)$  are successfully found. The patch function (25) with these polynomials as coefficients smoothly stitches together the two functions,  $f(r, \theta, \varphi)$  and  $g(r, \theta, \varphi)$ , along the  $r$  dimension on the interval  $[r_1, r_2]$ . Note that the method for stitching of two multivariable functions is not more complex than the method for stitching a single-variable function with a multivariable function from the previous Section.

So far we considered merging of the functions defined over real, three-dimensional space with the same set of independent variables. However, using patch function in the form given in (25) and the procedure described in this Section it is easy to show that the merged functions can have arbitrary numbers of variables, and even different numbers of variables. Having at least one variable in common is the only requirement.

Consider a more general example. Suppose two functions defined in eight-dimensional space,  $f(x, y, z, q, r)$  and  $g(q, r, s, t, u)$  need to be smoothly

merged along the  $q$  dimension over the interval  $[q_1, q_2]$ . The two functions have different sets of independent variables with non-empty intersection having  $q$  as its element. The patch function must be dependent on the union of the both sets of variables. It is defined in the form

$$p(x, y, z, q, r, t, u, s) = a(q)f(x, y, z, q_1, r) + b(r)g(q_2, r, s, t, u). \quad (33)$$

Following the reasoning from the previous Sections we can show that the cubic polynomials  $a(q)$  and  $b(q)$  can be determined through solving the linear systems (19) and (20) with the right hand side vectors

$$\mathbf{c}_a = \begin{bmatrix} 1; & 0; & \frac{f_q(x, y, z, q_1, r)}{f(x, y, z, q_1, r)}; & 0 \end{bmatrix}^T, \quad (34)$$

$$\mathbf{c}_b = \begin{bmatrix} 0; & 1; & 0; & \frac{g_q(q_2, r, s, t, u)}{g(q_2, r, s, t, u)} \end{bmatrix}^T. \quad (35)$$

## 6. Discussion

The method of stitching two functions along one dimension described here was developed in context of numerical simulations, which need to be optimized for computational efficiency and speed. If both chromospheric and coronal electron number density distributions depend only on the distance from the sun center  $r$ , the four coefficients of patch polynomial (7) can be calculated at the initialisation step. During the simulation there is no need to solve system (8); only the polynomial values need to be calculated. For example, in the ray tracing application this only happens when the ray enters the transition layer between the chromosphere and the corona. Therefore, the computational overhead of the stitching is relatively small.

In more complex cases of spherically non-symmetric coronal distributions like that by Saito (5), the linear system (20) must be solved at every step when the ray happens to travel inside the transition layer. This is required because the  $\mathbf{c}_b$  vector now depends on the  $\theta$  (and probably  $\varphi$ ) variables, whose values may change from one step to the next. Here are a few tips to accelerate the computations. The fact that both linear systems have the same system matrix  $\mathbf{P}$  (10), is one of the benefits of the method.  $\mathbf{P}$  only depends on the values of  $r_1$  and  $r_2$  (i.e. boundaries of the transition layer), which are usually held constant

during the course of computations. This makes it possible to calculate  $\mathbf{P}$  once and for all before the actual raytracing starts. To save even more time one can replace the linear system solving with matrix-vector multiplying. To do so,  $\mathbf{P}^{-1}$ , must be calculated during the initialization phase. The vector  $\mathbf{a}$  of polynomial (21) coefficients must also be pre-calculated as the product of the matrix  $\mathbf{P}^{-1}$  and the vector  $\mathbf{c}_a$ :

$$\mathbf{a} = \mathbf{P}^{-1}\mathbf{c}_a. \quad (36)$$

Then, in the process of ray tracing, only the coefficients of the polynomial  $b(r)$  (see (22)) must be calculated as

$$\mathbf{b} = \mathbf{P}^{-1}\mathbf{c}_b, \quad (37)$$

and followed by computing the patch function (11) itself. Adopting this scheme makes computations significantly faster. Note, however, that in case of stitching two three-variable functions both operations, (36) and (37), must be repeated at every ray tracing step.

In the previous Section it is shown that this approach to stitching multi-variate functions is not limited to the three-dimensional cases. It is general enough to be applied to abstract functions of any number of independent variables, while retaining its simplicity. For any number of dimensions only eight parameters, the coefficients of the  $3^{rd}$  order polynomials  $a(r)$  and  $b(r)$  need to be determined.

## 7. Conclusion

The method developed here is intended for smooth merging of the electron number density model distributions at the boundary between the solar corona and chromosphere. Used extensively in our coronal radio propagation studies, the method has shown excellent results. We have also shown that the method can easily be generalized to the problems where two abstract functions of arbitrary variable numbers need to be smoothly stitched along one common dimension.

## Acknowledgements

This work was partially supported under grants from the National Science Foundation, Division of Astronomical Sciences and Atmospheric and Geospace Sciences to the MIT Haystack Observatory.

## REFERENCES

- Oberoi, D. et al., 2011, First spectroscopic imaging observations of the sun at low radio frequencies with the Murchison Widefield Array prototype, *ApJ*, **728**, L27
- Benkevitch, L., Sokolov, I., Oberoi, D., & Zurbuchen, T. 2010, Algorithm for tracing radio rays in solar corona and chromosphere, arXiv:1006.5635v3
- Lonsdale, C. J. et al., 2009, The Murchison Widefield Array: design overview, *Proc. IEEE*, **97**, 1497
- Ellingson, S. W. et al., 2009, The Long Wavelength Array, *Proc IEEE*, **97**, 1421
- de Vos, M., Gunst, A. W., Nijboer, R., 2009, The LOFAR telescope: system architecture and signal processing, *Proc. IEEE*, **97**, 1431
- K. Saito, 1970, A non-spherical axisymmetric model of the solar K corona of the minimum type, *Ann. Tokyo Astron. Obs.*, Second Ser., vol. **12**, 53-120
- G. Newkirk, Jr., 1961, The solar corona in active regions and the thermal origin of the slowly varying component of solar radio radiation, *ApJ*, **133**, 983
- C. W. Allen, 1947, Interpretation of electron densities from corona brightness, *Mon. Not. R. Astr. Soc.*, vol. **107**, 426
- Cillie, G. G., and Menzel, D. H., 1935, The physical state of the solar chromosphere, *Harv. Coll. Obs. Circ.*, **410**, 1-40